

类的抽象过程、构造方法

- ◆ 面向对象
- ◆ 用面向对象描述世界
- ◆ 构造方法
- ◆ 带参构造方法与方法的重载

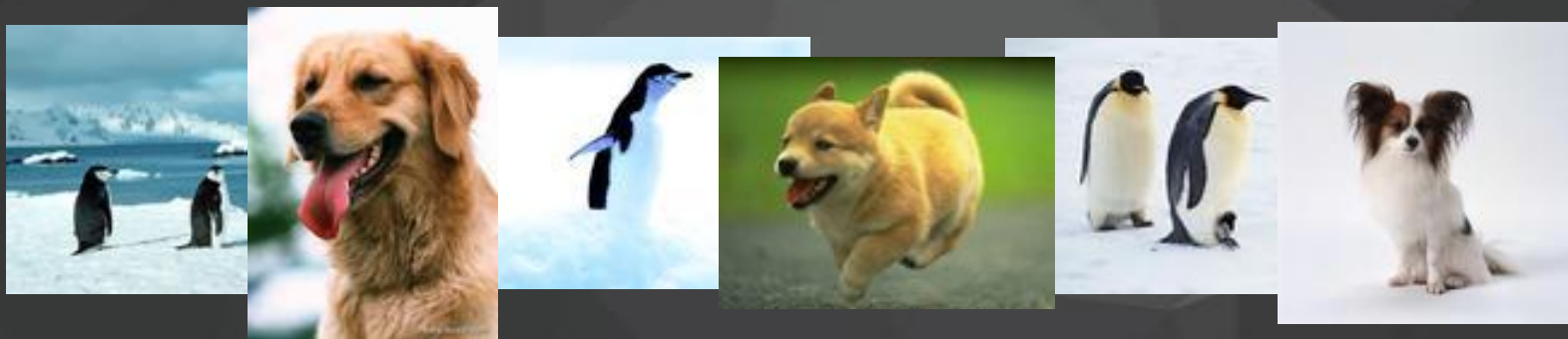
现实世界是由什么组成的？



世界由对象组成

◀ 一个现实世界的问题

▶ 宠物——现实世界的对象



▶ 如何在计算机中描述它？

从现实中抽象出类分三步：

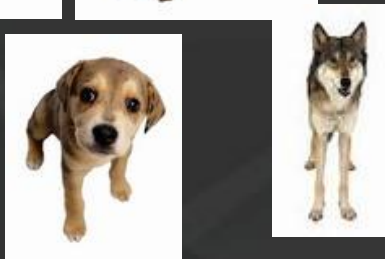
1. 找出它的种类
2. 找出它的属性
3. 找出它的行为

◆ 用面向对象描述世界

▶ 用面向对象的思想描述世界

▶ 第一步：发现类

各个狗对象



名词

```
class Dog {  
      
}
```

根据“对象”抽象出“类”

◆ 用面向对象描述世界

▶ 第二步：发现类的属性

名词

狗类共有的特征：

1. 品种
 2. 年龄
 3. 昵称
 4. 健康情况
 5. 跟主人的亲密度
-

只放我关心的
属性

```
class Dog {  
    String name = "旺财"; // 昵称  
    int health = 100; // 健康值  
    int love = 0; // 亲密度  
    String strain = "拉布拉多犬"; // 品种  
}
```

用面向对象描述世界

第三步：发现类的方法

动词

狗类共有的行为：

1. 跑
2. 吠
3. 输出狗的信息

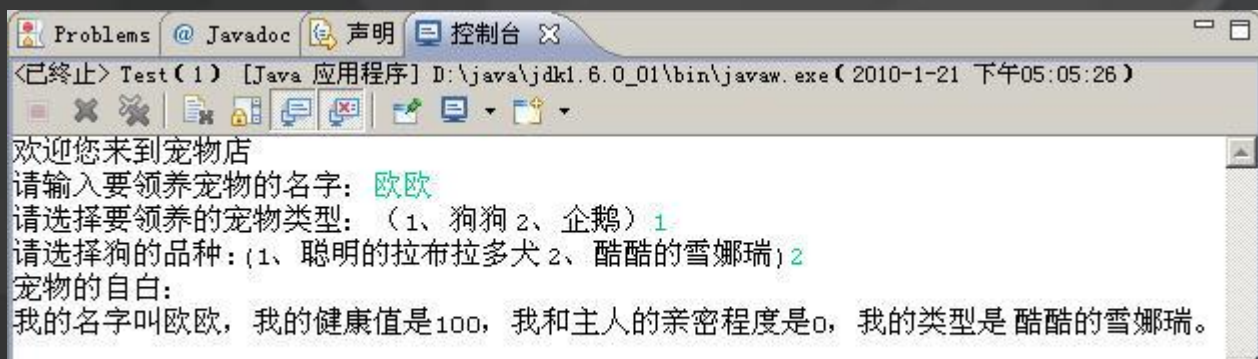
... ..

只放我关心的
方法

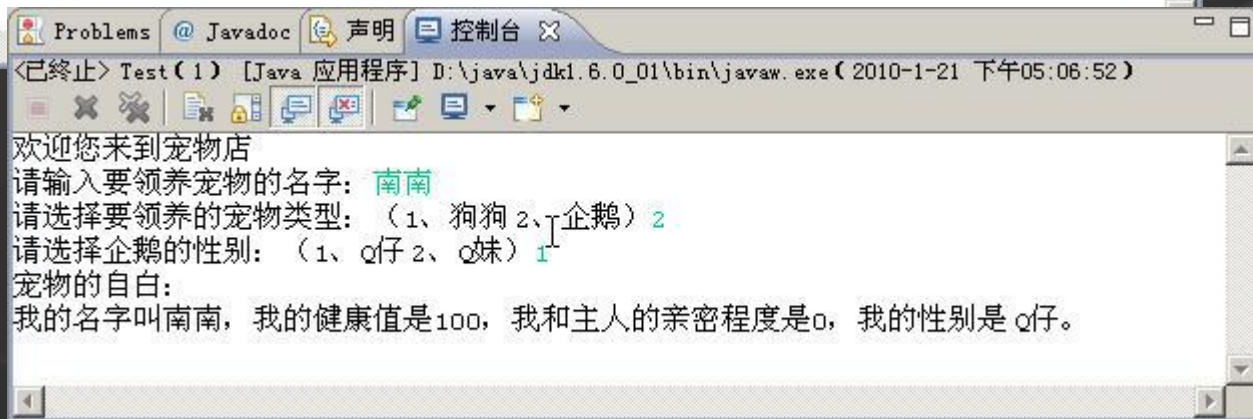
```
class Dog {  
    String name = "旺财"; // 昵称  
    int health = 100; // 健康值  
    int love = 0; // 亲密度  
    String strain = "拉布拉多犬"; // 品种  
    /* 输出狗的信息 */  
    public void print() {  
        // 输出狗信息的代码  
    }  
}
```


实现领养宠物

- ▶ 实现领养宠物功能
 - ▶ 编写Dog类和Penguin类
 - ▶ 创建宠物对象，输入宠物信息，最后输出



```
Problems @ Javadoc 声明 控制台 X
<已终止> Test (1) [Java 应用程序] D:\java\jdk1.6.0_01\bin\javaw.exe (2010-1-21 下午05:05:26)
欢迎您来到宠物店
请输入要领养宠物的名字: 欧欧
请选择要领养的宠物类型: (1、狗狗 2、企鹅) 1
请选择狗的品种: (1、聪明的拉布拉多犬 2、酷酷的雪娜瑞) 2
宠物的自白:
我的名字叫欧欧, 我的健康值是100, 我和主人的亲密程度是0, 我的类型是 酷酷的雪娜瑞。
```



```
Problems @ Javadoc 声明 控制台 X
<已终止> Test (1) [Java 应用程序] D:\java\jdk1.6.0_01\bin\javaw.exe (2010-1-21 下午05:06:52)
欢迎您来到宠物店
请输入要领养宠物的名字: 南南
请选择要领养的宠物类型: (1、狗狗 2、企鹅) 2
请选择企鹅的性别: (1、Q仔 2、Q妹) 1
宠物的自白:
我的名字叫南南, 我的健康值是100, 我和主人的亲密程度是0, 我的性别是 Q仔。
```


对象初始化

使用构造方法：

```
Penguin pgn1 = new Penguin();
```

```
pgn.sex
```

```
class Penguin {
```

```
// 属性
```

```
/* 无参
```

```
public P
```

```
name = "qq";
```

```
love = 20;
```

```
sex = "Q仔";
```

```
System.out.println("执行构造方法");
```

```
}
```

```
}
```

能否在创建对象 构造方法
完成赋值?

构造方法

没有返回值
类型

与类名相同

```
public 构造方法名() {  
    //初始化代码  
}
```

可以指定
参数

```
public Penguin() {  
    name = "qq";  
    love = 20;  
    sex = "Q仔";  
    System.out.println("执行构造方法");  
}
```

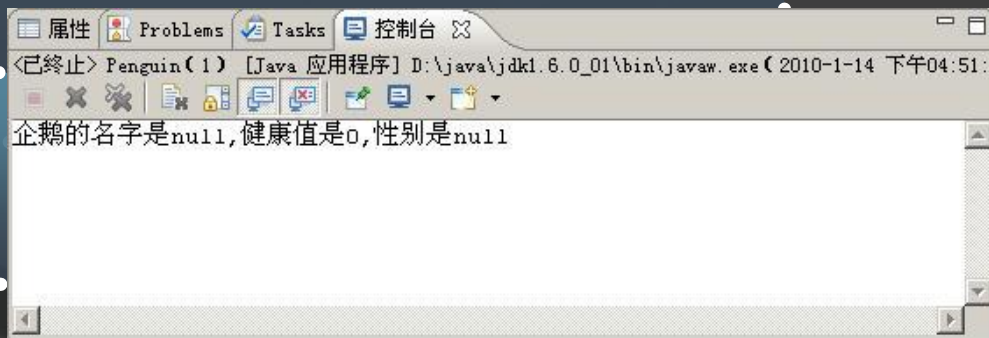
构造方法

▶ 阅读代码，说出运行结果，并指出原因

```
class Penguin {  
    public void Penguin() {  
        health=10;  
        sex="雄";  
        System.out.println("执行构造方法");  
    }  
    public void print() {  
        System.out.println("企鹅的名字是" + name  
            + ",健康值是" + health  
            + ",性别是" + sex);  
    }  
}
```

有返回值类型，
不是构造方法

```
Penguin pgn3=new Penguin();  
pgn3.print();
```



◆ 带参构造方法与方法的重载

- ▶ 一个类是否可以有多个构造方法？
- ▶ 方法重载，指同一个类中多个方法：
 - ▶ 方法名相同
 - ▶ 参数列表不同

与返回值、访问修饰符无关

```
System.out.println(45);  
System.out.println(true);  
System.out.println("狗在玩耍！");
```

示例1

```
public Penguin() {  
    //代码  
}  
public Penguin(String name, int health, int love, String sex) {  
    //代码  
}
```

示例2

- ▶ 已有Student类，包含的属性
 - ▶ 学号
 - ▶ 姓名
 - ▶ 年龄
- ▶ 创建学生对象时，支持以下构造方式
 - ▶ 构造时需提供姓名、年龄
 - ▶ 学号设为默认值 “0000”

理解方法重载

生活中的方法重载

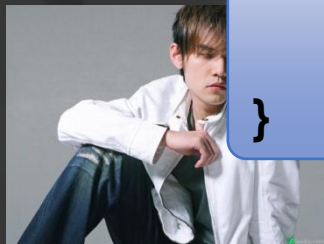
类

方法

参数

方法体

```
public class Player{  
    public void play(剧本){ //根据剧本表演}  
    public void play(歌曲){ //演唱歌曲}  
    public void play(钢琴){ //弹奏钢琴}  
}
```



表演

歌曲



方法重载：同一个类中，方法名相同，参数项不同
(参数类型不同、参数个数不同)



► 阅读以下代码，判断哪些属于方法重载？

1) `public void study(String name){}`
`public void study(int hours){}`

方法重载

2) `public void study(String name){}`
`public string study(String name){}`

3) `public void study(String name){}`
`public boolean study(String name, int age){}`

方法重载

4) `public student(String name){}`
`public student(String name, int age){}`

构造函数重载

方法重载有两种：
参数类型不同的重载、参数个数不同的重载

▶ 编写Add类

- ▶ 添加一个方法sum()，返回类型int，有两个int类型参数。该方法计算两个整数的和
- ▶ 添加sum()方法，返回类型double，有两个double类型的参数。该方法计算两个double数据的和
- ▶ 添加sum()方法，返回类型String，有两个String类型的参数。该方法将两个字符串拼接。

谢 谢

